



**SGCP's**

**Guru Nanak Khalsa College**

**of Arts, Science & Commerce (Autonomous)**

**Mini-Project:**

**Bookshop Management Program**

Team Member's Roll Number	Team Member's Name
102	Bhalekar Esha Gajanan Samruddhi
110	Kasturi Nayan Prabhakar Anuradha
115	Kothari Prarthi Hrishit Dhara

**COURSE: M. Sc. Bioinformatics - Part I**

**ACADEMIC YEAR: 2023-2024**

**PAPER: Mandatory Paper II**

**PAPER CODE: GNKPSBI2501**

**COURSE TITLE:**

**Introduction to Programming Languages and Databases**

## **AIM OF THE PROJECT:**

The aim of the project is to develop a console-based bookstore management system using C++ programming language that utilizes Object-Oriented Programming (OOP) principles to perform relevant calculations and facilitate the billing process for the small-sized bookshop.

## **OBJECTIVE OF THE PROJECT:**

The Bookstore Management program is designed with several key objectives in mind. The objectives of the Bookstore Management program are as follows:

1. Customer information management:
  - To manage customer data, including their names and contact numbers.
  - To ensure data integrity by validating customer name and phone number inputs.
2. Comfort & ease to the cashiers:
  - To allow cashiers to select multiple books from a predefined list using book IDs and calculate the total price of selected books, along with printing of the bill.
  - To implement efficient book selection and billing processes to simplify the overall billing experience for cashiers.
3. Error handling:
  - Handle input errors for customer details, book selection as well as for the history of transactions.
4. Order processing and order ID generation:
  - To generate a unique order ID for each transaction.
  - To record and save order details, including the order ID, customer's name, contact number, and the total order amount.
  - To provide a summary of the final order.
5. Billing and documentation:
  - To produce a bill with essential information, such as the order ID, customer details, detailed list of purchased books as well as total order price.
  - To save transaction records to a file for future reference.
  - To generate a printable bill for the customer, in .txt format.

6. User-friendly interface:

- To create a user-friendly console interface with easy interactions by prompting users for necessary inputs and providing clear instructions throughout the process.

7. Robust system:

- To enable the system to handle multiple transactions continuously until instructed to exit.

These objectives collectively aim to provide a well-structured and efficient system for managing a small-sized bookshop's operations, enhancing customer service, and maintaining organized records for future analysis and reference.

## Code:

```
#include <iostream> // for basic input output
#include <string> // storing characters in 1D array and string manipulation
#include <regex> //setting regular expression and verification of user input
#include <fstream> // for creating, modifying and writing the files
#include <ctime> // for accessing date and time
using namespace std;

string books[][4] = { //books array
    {"1", "It Ends With Us", "(Collen Hoover)", "100"},
    {"2", "Milk and Honey", "(Rupi Kaur)", "200"},
    {"3", "Call Me by Your Name", "(Andre Aciman)", "300"},
    {"4", "Nexus Sentinels", "(Nayan Kasturi)", "400"},
    {"5", "Verity", "(Collen Hoover)", "500"},
    {"6", "That Sapphire Night", "(Swastika Jha)", "600"},
    {"7", "Sorry You're Not My Type", "(Sudeep Nagarkar)", "700"},
    {"8", "The Fault In Our Stars", "(John Green)", "800"},
    {"9", "After", "(Anna Todd)", "900"},
    {"10", "Healing Words", "(Alexandra Vasiliu)", "1000"}
};

class CustomerDetails { // stores and returns customer details
private:
    string name;
    string num;

public:
    string cname() { // takes customer's name and verifies it
        regex pattern("^[a-zA-Z ]+$");
        bool valid = false;
        while (!valid) {
            cout << "Enter customer's name: ";
            cin >> ws; //removes the whitespace from previous execution of
int main()

            getline(cin, name);

            if (regex_match(name, pattern)) {
                valid = true;
            } else {
                cout << "Invalid name. Please enter a valid customer's
name." << endl;
            }
        }
        cout << "Customer's Name: " << name << endl;
        return name;
    }
}
```

```

        string cnum() { // takes customer's contact number and verifies it
            regex pattern("(0|91)?[6-9][0-9]{9}");
            bool valid = false;
            while (!valid) {
                cout << "Enter customer's contact number: ";
                cin >> num;

                if (regex_match(num, pattern)) {
                    valid = true;
                } else {
                    cout << "Invalid contact number. Please enter a valid
customer's contact number." << endl;
                }
            }

            cout << "Customer's Contact Number: " << num << endl;
            return num;
        }
    };

class OrderCalc: public CustomerDetails { // child class containing order
processing function
private:
    int total_price;

public:
    int orderProcess(string purchased[][4], int &bookCount) { // order
processing and returns the total order amount
        bool flag = true;
        total_price = 0;

        cout << "Enter 0 to exit the program" << endl;
        cout << "Enter book ID: ";

        while (flag) {
            int bookID;
            cin >> bookID;

            if (bookID == 0) {
                flag = false;
                break;
            } else if (bookID >= 1 && bookID <= 10 && bookCount < 100) {
                purchased[bookCount][0] = to_string(bookID);
                purchased[bookCount][1] = books[bookID - 1][1];
                purchased[bookCount][2] = books[bookID - 1][2];
                purchased[bookCount][3] = books[bookID - 1][3];
            }
        }
    }
};

```

```

        for (int a = 0; a < 4; a++) {
            cout << purchased[bookCount][a] << "\t";
        }

        int book_price_converted = stoi(purchased[bookCount][3]);
        total_price += book_price_converted;
        bookCount++;
    } else {
        cout << "\nInvalid Book ID. Please enter a number between
1 and 10." << endl;
    }

    cout << "\nEnter 0 to exit the program";
    cout << "\nEnter next Book ID for further purchase: ";
}

cout << "\nFinal List of books purchased: " << endl;
for (int j = 0; j < bookCount; j++) {
    for (int a = 0; a < 4; a++) {
        cout << purchased[j][a] << "\t";
    }
    cout << endl;
}

cout << "-----" << endl;
cout << "                Total Price : ₹" << total_price <<
endl;
cout << "-----" << endl;

return total_price;
}
};

class Transactions: public OrderCalc { // child class storing transaction
history and creating bills and files
public:
    string generateOrderID() { // generates serial - wise order IDs
        static int orderNumber = 1; // Static variable to keep track of
order numbers
        string orderID = "ABOID_" + to_string(orderNumber);
        orderNumber++;
        return orderID;
    }
}

```

```

        void writeTXT (string orderID, string custname, string custnum, int
totalPrice){ // creates and writes the transaction in .txt file
        fstream f;
        f.open("transactions.txt", ios::app);
        if (!f) {
            ofstream MyFile("transactions.txt");
            f << orderID << "\t\t\t" << custname << "\t\t\t" << custnum <<
"\t\t\t\t" << totalPrice << "\n";
            cout << "Bill Printed Successfully" << endl;
            f.close();
        }
        else {
            f << orderID << "\t\t\t" << custname << "\t\t\t" << custnum <<
"\t\t\t\t" << totalPrice << "\n";
            cout << "Bill Printed Successfully" << endl;
            f.close();
        }
    }

    void saveTransaction(string orderID, string custname, string custnum,
int totalPrice, string purchased[][4], int bookCount) { // creates a formatted
bill in console
        writeTXT(orderID, custname, custnum, totalPrice);
        cout << endl;
        cout << "#####" <<
endl;
        cout << "                ARK Book Store                " <<
endl;
        cout << "-----" <<
endl;
        cout << endl;
        cout << "Bill No.: " << orderID << endl;
        cout << "Customer Name: " << custname << endl;
        cout << "Customer Number: " << custnum << endl;
        cout << endl;
        cout << "-----" <<
endl;
        cout << "                Purchased Books                " <<
endl;
        cout << "-----" <<
endl;
        cout << endl;
        for (int i = 0; i < bookCount; i++) {
            for (int a = 0; a < 4; a++) {
                cout << purchased[i][a] << "\t";
            }
        }
    }

```

```

        cout << endl;
    }
    cout << endl;
    cout << "-----" <<
endl;
    cout << "                                Total: ₹" << totalPrice <<
endl;
    cout << "#####" <<
endl;
    cout << endl;
}

void printBill(string custname, string custnum, string orderID, string
purchased[100][4], int totalPrice, int bookCount){
    // creates a formatted bill in .txt file
    fstream f;
    string billNo = orderID + ".txt";
    ofstream MyFile(billNo);
    f.open(billNo);
    f << endl;
    f << "#####" << endl;
    f << "                                ARK Book Store                                " << endl;
    f << "-----" << endl;
    f << endl;
    f << "Bill No.: " << orderID << endl;
    f << "Customer Name: " << custname << endl;
    f << "Customer Number: " << custnum << endl;
    f << endl;
    f << "-----" << endl;
    f << "                                Purchased Books                                " << endl;
    f << "-----" << endl;
    f << endl;
    for (int i = 0; i < bookCount; i++) {
        for (int a = 0; a < 4; a++) {
            f << purchased[i][a] << "\t\t";
        }
        f << endl;
    }
    f << endl;
    f << "-----" << endl;
    f << "                                Total: ₹" << totalPrice << endl;
    f << "#####" << endl;
    f << endl;
    f.close();
}
};

```



```

int main() {

    while (true) { // loop - ensures continuous program execution
        string custname, custnum, orderID;
        string purchased[100][4];
        int totalPrice, bookCount = 0;

        Transactions clt;
        cout << "### BOOKSTORE MANAGEMENT SYSTEM by EB, NK & PK ###" << endl;

        custname = clt.cname(); // takes in customer's name and verifies it
        custnum = clt.cnum(); // takes in customer's contact number and
verifies it
        totalPrice = clt.orderProcess(purchased, bookCount); // takes in book
IDs and returns the calculated total price
        orderID = clt.generateOrderID(); // generates sequential order IDs
        clt.saveTransaction(orderID, custname, custnum, totalPrice, purchased,
bookCount); // prints the formatted bill in console
        clt.printBill(custname, custnum, orderID, purchased, totalPrice,
bookCount); // creates .txt file w.r.t. bill number and writes formatted bill
in it

        cout << "Enter 0 to terminate program or any other key to continue: ";
        string choice;
        cin >> choice;
        if (choice == "0") {
            cout << endl;
            break; // Exits the loop and interrupts the program execution, if
the user enters 0
        }
        else{
            cout << endl;
            continue;
        }
    }
    return 0;
}

```

## OUTPUT:

### 1. Code execution

```
### BOOKSTORE MANAGEMENT SYSTEM by EB, NK & PK ###
Enter customer's name: John Doe
Customer's Name: John Doe
Enter customer's contact number: 8767932277
Customer's Contact Number: 8767932277
Enter 0 to exit the program
Enter book ID: 2
2      Milk and Honey  (Rupi Kaur)      200
Enter 0 to exit the program
Enter next Book ID for further purchase: 4
4      Nexus Sentinels (Nayan Kasturi) 400
Enter 0 to exit the program
Enter next Book ID for further purchase: 6
6      That Sapphire Night (Swastika Jha) 600
Enter 0 to exit the program
Enter next Book ID for further purchase: 5
5      Verity (Collen Hoover) 500
Enter 0 to exit the program
Enter next Book ID for further purchase: 8
8      The Fault In Our Stars (John Green) 800
Enter 0 to exit the program
Enter next Book ID for further purchase: 9
9      After (Anna Todd) 900
Enter 0 to exit the program
Enter next Book ID for further purchase: 1
1      It Ends With Us (Collen Hoover) 100
Enter 0 to exit the program
Enter next Book ID for further purchase: 0

Final List of books purchased:
2      Milk and Honey  (Rupi Kaur)      200
4      Nexus Sentinels (Nayan Kasturi) 400
6      That Sapphire Night (Swastika Jha) 600
5      Verity (Collen Hoover) 500
8      The Fault In Our Stars (John Green) 800
9      After (Anna Todd) 900
1      It Ends With Us (Collen Hoover) 100
-----
                        Total Price : ₹3500
-----
Bill Printed Successfully
```

Fig 1.1: Input of Customer Details & purchased books,  
with output of order processing and calculation of total order price

```

-----
Bill Printed Successfully

#####
                ARK Book Store
-----

Bill No.: AB0ID_1
Customer Name: John Doe
Customer Number: 8767932277

-----
                Purchased Books
-----

2      Milk and Honey  (Rupi Kaur)      200
4      Nexus Sentinels (Nayan Kasturi) 400
6      That Sapphire Night (Swastika Jha) 600
5      Verity (Collen Hoover) 500
8      The Fault In Our Stars (John Green) 800
9      After (Anna Todd) 900
1      It Ends With Us (Collen Hoover) 100

-----
                        Total: ₹3500
#####

Enter 0 to terminate program or any other key to continue: p

```

Fig 1.2: Output of formatted order bill in the console/terminal

```

Enter 0 to terminate program or any other key to continue: p

### BOOKSTORE MANAGEMENT SYSTEM by EB, NK & PK ###
Enter customer's name: Natasha Romanoff
Customer's Name: Natasha Romanoff
Enter customer's contact number: 9767287799
Customer's Contact Number: 9767287799
Enter 0 to exit the program
Enter book ID: 5
5      Verity (Collen Hoover) 500
Enter 0 to exit the program
Enter next Book ID for further purchase: 6
6      That Sapphire Night (Swastika Jha) 600
Enter 0 to exit the program
Enter next Book ID for further purchase: 2
2      Milk and Honey (Rupi Kaur) 200
Enter 0 to exit the program
Enter next Book ID for further purchase: 7
7      Sorry You're Not My Type (Sudeep Nagarkar) 700
Enter 0 to exit the program
Enter next Book ID for further purchase: 8
8      The Fault In Our Stars (John Green) 800
Enter 0 to exit the program
Enter next Book ID for further purchase: 0

Final List of books purchased:
5      Verity (Collen Hoover) 500
6      That Sapphire Night (Swastika Jha) 600
2      Milk and Honey (Rupi Kaur) 200
7      Sorry You're Not My Type (Sudeep Nagarkar) 700
8      The Fault In Our Stars (John Green) 800
-----
Total Price : ₹2800
-----
Bill Printed Successfully

```

Fig 1.3: Continuous execution of the program,  
with a choice for user to terminate program execution

```
-----  
Bill Printed Successfully
```

```
#####
```

```
ARK Book Store
```

```
-----  
Bill No.: ABOID_2
```

```
Customer Name: Natasha Romanoff
```

```
Customer Number: 9767287799
```

```
-----  
Purchased Books
```

```
-----  
5      Verity (Collen Hoover) 500  
6      That Sapphire Night (Swastika Jha) 600  
2      Milk and Honey (Rupi Kaur) 200  
7      Sorry You're Not My Type (Sudeep Nagarkar) 700  
8      The Fault In Our Stars (John Green) 800
```

```
-----  
Total: ₹2800
```

```
#####
```

```
Enter 0 to terminate program or any other key to continue: 0
```

```
PS D:\bookstore-manager> █
```

Fig 1.4: Termination of the program,  
with a choice for user to continue program execution

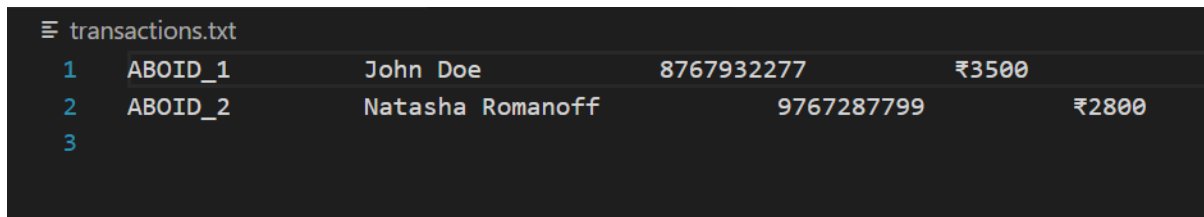
## 2. File handling and manipulation

```
ABOID_1.txt
1
2 #####
3 | | | | | ARK Book Store
4 -----
5
6 Bill No.: ABOID_1
7 Customer Name: John Doe
8 Customer Number: 8767932277
9
10 -----
11 | | | | | Purchased Books
12 -----
13
14 2      Milk and Honey      (Rupi Kaur)      200
15 4      Nexus Sentinels     (Nayan Kasturi)    400
16 6      That Sapphire Night (Swastika Jha)      600
17 5      Verity              (Collen Hoover)    500
18 8      The Fault In Our Stars (John Green)    800
19 9      After               (Anna Todd)       900
20 1      It Ends With Us     (Collen Hoover)    100
21
22 -----
23 | | | | | Total: ₹3500
24 #####
25

ABOID_2.txt
1
2 #####
3 | | | | | ARK Book Store
4 -----
5
6 Bill No.: ABOID_2
7 Customer Name: Natasha Romanoff
8 Customer Number: 9767287799
9
10 -----
11 | | | | | Purchased Books
12 -----
13
14 5      Verity              (Collen Hoover)    500
15 6      That Sapphire Night (Swastika Jha)      600
16 2      Milk and Honey      (Rupi Kaur)      200
17 7      Sorry You're Not My Type (Sudeep Nagarkar) 700
18 8      The Fault In Our Stars (John Green)    800
19
20 -----
21 | | | | | Total: ₹2800
22 #####
23
```

Fig 2.1: Creation of individual TXT files with respect to order ID and writing the formatted and printable bill in the created TXT files

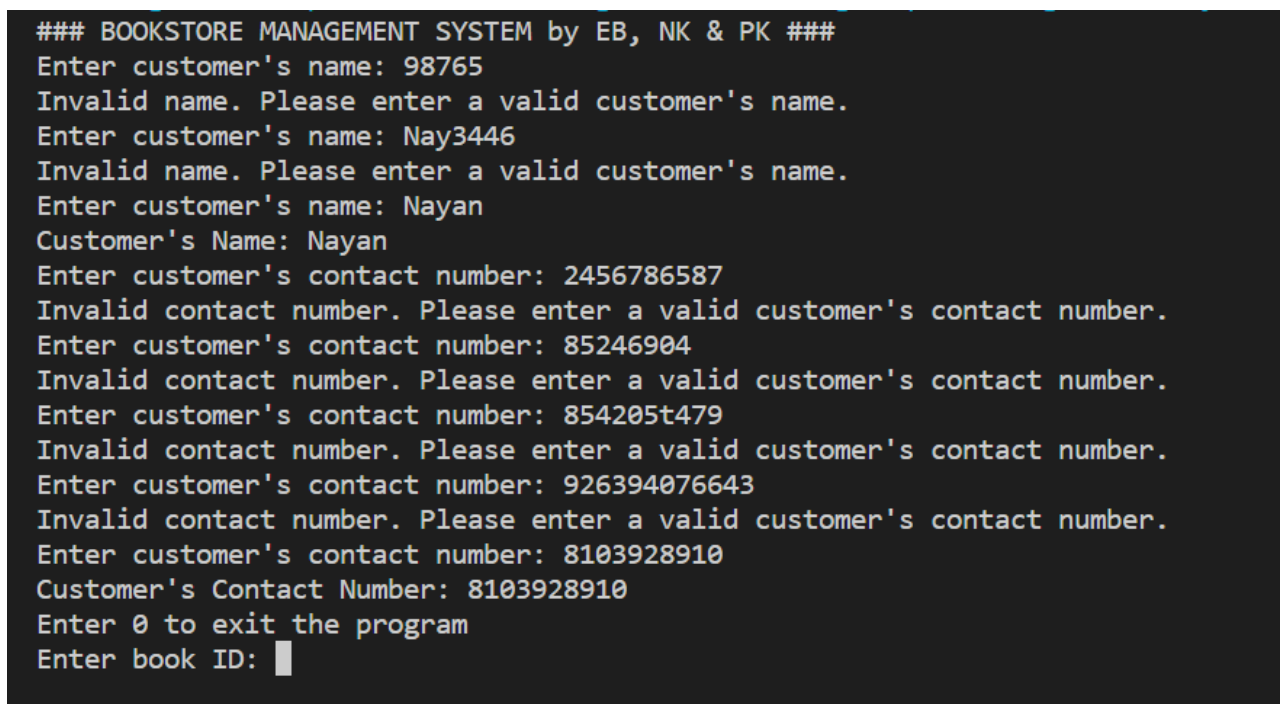
Legend:	ABOID_1: Formatted bill of first transaction
	ABOID_2: Formatted bill of second transaction



1	AB0ID_1	John Doe	8767932277	₹3500
2	AB0ID_2	Natasha Romanoff	9767287799	₹2800
3				

Fig 2.2: Creation of tranactions.txt file and writing the transaction details like, Order ID, Customer Details and Total Amount of Order in created TXT file

### 3. Error handling



```

### BOOKSTORE MANAGEMENT SYSTEM by EB, NK & PK ###
Enter customer's name: 98765
Invalid name. Please enter a valid customer's name.
Enter customer's name: Nay3446
Invalid name. Please enter a valid customer's name.
Enter customer's name: Nayan
Customer's Name: Nayan
Enter customer's contact number: 2456786587
Invalid contact number. Please enter a valid customer's contact number.
Enter customer's contact number: 85246904
Invalid contact number. Please enter a valid customer's contact number.
Enter customer's contact number: 854205t479
Invalid contact number. Please enter a valid customer's contact number.
Enter customer's contact number: 926394076643
Invalid contact number. Please enter a valid customer's contact number.
Enter customer's contact number: 8103928910
Customer's Contact Number: 8103928910
Enter 0 to exit the program
Enter book ID: █

```

Fig 3.1: Verification of customer details and prompting for the input of valid details

```

### BOOKSTORE MANAGEMENT SYSTEM by EB, NK & PK ###
Enter customer's name: John Doe
Customer's Name: John Doe
Enter customer's contact number: 9458762039
Customer's Contact Number: 9458762039
Enter 0 to exit the program
Enter book ID: 18

Invalid Book ID. Please enter a number between 1 and 10.

Enter 0 to exit the program
Enter next Book ID for further purchase: 3
3      Call Me by Your Name      (Andre Aciman)  300
Enter 0 to exit the program
Enter next Book ID for further purchase: 10
10     Healing Words      (Alexandra Vasiliu)    1000
Enter 0 to exit the program
Enter next Book ID for further purchase: 0

Final List of books purchased:
3      Call Me by Your Name      (Andre Aciman)  300
10     Healing Words      (Alexandra Vasiliu)    1000
-----
                        Total Price : ₹1300
-----
Bill Printed Successfully

```

Fig 3.2: Verification of book ID and prompting for the valid input details before further execution

## **REFERENCES:**

1. Kirch-Prinz, U., & Prinz, P. (2002, January 1). *A Complete Guide to Programming in C++*. Jones & Bartlett Learning.
2. Schildt, H. (2002, November 19). *C++: The Complete Reference, 4th Edition*. McGraw Hill Professional.
3. *File Handling through C Classes*. (2023, April 18). GeeksforGeeks. <https://www.geeksforgeeks.org/file-handling-c-classes/>
4. *C Program to Print a 2D Array*. (2022, July 19). GeeksforGeeks. <https://www.geeksforgeeks.org/cpp-program-to-print-a-2d-array/>

## **GitHub Repository Link:**

Bookstore Manager - <https://github.com/RaannaKasturi/Bookstore-Manager>